AUTOMATIC TEST SYSTEMS SUBDOMAIN ANNEX FOR THE COMBAT SUPPORT DOMAIN

CS.ATS.1 SUBDOMAIN OVERVIEW

CS.ATS.1.1 PURPOSE

The Automatic Test Systems (ATS) Subdomain Annex identifies additions to the Combat Support Domain Annex core elements (i.e., standards, interfaces, and service areas) listed in Section 2 of this document. These additions are common to the majority of ATS and support the functional requirements of these systems.

The purpose of the ATS Subdomain Annex is:

– To provide the foundation for a seamless flow of information and interoperability among all Department of Defense (DoD) ATS.
– To mandate standards and guidelines for system development and acquisition which will significantly reduce cost, development time, and fielding time for improved systems, while minimizing the impact on program performance wherever possible.
– To improve the test acquisition process by creating an ATS framework that can meet functional and technical needs, promote automation in software development, re-hostability and portability of Test Program Sets (TPSs).
– To communicate to industry DoD’s intention to use open systems products and implementations. DoD will buy commercial products and systems, which use open standards, to obtain the most value for limited procurement dollars.

CS.ATS.1.2 BACKGROUND

From 1980 to 1992, the US DoD investment in depot and factory ATS exceeded $35 billion with an additional $15 billion for associated support. Often, application specific test capability was procured by weapon systems acquisition offices with little coordination among DoD offices. This resulted in a proliferation of different custom equipment types with unique interfaces that made the DoD appear to be a variety of separate customers. To address this problem, the DoD enacted policy changes that require that *“Automatic Test System capabilities be defined through critical hardware and software elements.”* In response, the joint service Automatic Test Systems (ATS) Research and Development (R&D) Integrated Product Team (IPT) (ARI) sponsored the Critical Interfaces (CI) Working Group, which recommended interfaces and standards that should be mandated for DoD ATS acquisitions. The CI report became the basis for this document which is an annex to the Joint Technical Architecture (JTA). The ATS Subdomain Annex will aid in satisfying the requirements of DoD Regulation 5000.2-R to migrate DoD designated tester families towards a common architecture.

The policy changes listed below require DoD offices to take a unified corporate approach to acquisition of ATS.

– DoD Regulation 5000.2-R, "Mandatory Procedures for Major Defense Acquisition Programs and Major Automated Information System Acquisition Programs", March 15, 1996[1], brings a cost-effective approach to the acquisition of ATS. This policy requires hardware and software needs for depot and intermediate-level applications to be met using DoD designated families and commercial equipment with defined interfaces and requires the management of ATS as a separate commodity through a DoD Executive Agent Office (EAO).
– Secretary of Defense Memorandum on Specifications and Standards - 29 June 1994, directs that DoD procurements will be made first by performance definition, second by commercial standards, and finally (and only with waiver) by military standards.

The use of open standards in ATS has been projected to provide the following five benefits.[2]

– Improve the test acquisition process by creating an ATS framework that can meet functional and technological needs, and promote automation in software development, re-hostability, and portability of Test Program Sets (TPSs).
– Decrease the use of custom hardware from approximately 70% today to 30%.
– Reduce engineering costs 70%.
– Reduce TPS integration time and cost 50-75%.
– Provide an iterative improvement in the quality of test by the reuse and refinement of libraries.

CS.ATS.1.3 SUBDOMAIN DESCRIPTION

A high level overview of a typical ATS is shown in Figure CS.ATS-1. An ATS has three major components: Automated Test Equipment (ATE), TPSs, and the Test Environment. The ATE consists of test and measurement instruments, a host computer, switching, communication busses, a receiver, and system software. The host computer controls the test and measurement equipment and execution of the TPS. The system software controls the test station and allows TPSs to be developed and executed. Examples of system software include operating systems, compilers, and test executives. The TPS consists of software to diagnose Units Under Test (UUTs), a hardware fixture that connects the UUT to the ATE, and documentation that instructs the station operator how to load and execute the TPS. The Test Environment includes a description of the ATS Architecture, programming and test specification languages, compilers, development tools, and a standard format for describing UUT design requirements and test strategy information that allows TPS software to be produced at a lower cost. The ATS architecture shown in Figure CS.ATS-1 is expanded into more detail in the hardware and software technical reference models introduced in Section CS.ATS.1.4. Each interface in the technical reference models is discussed in more detail in Sections CS.ATS.2 and CS.ATS.3.

---

[1] DoD Regulation 5000.2-R, paragraph 4.3.3.4, 15 March 1996. *"DoD Automated Test System (ATS) families or COTS components that meet defined ATS capabilities shall be used to meet all acquisition needs for automatic test equipment hardware and software. ATS capabilities shall be defined through critical hardware and software elements. The introduction of unique types of ATS into the DoD field, depot, and manufacturing operations shall be minimized."*

[2] Institute for Defense Analysis (IDA) *Investment Strategy Study* 1993

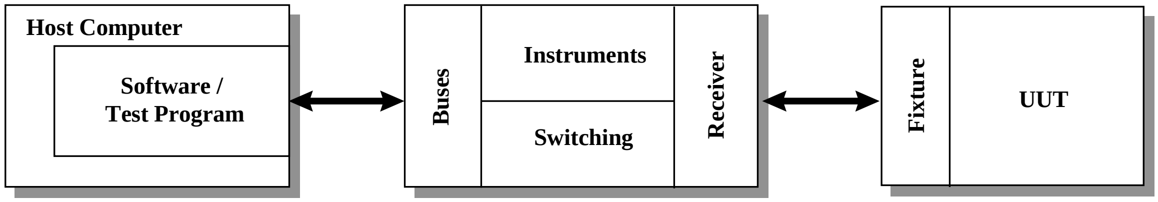

Figure CS.ATS-1 Generic ATS Architecture

CS.ATS.1.4 SCOPE AND APPLICABILITY

The following factors guided the selection of interfaces in the ATS Subdomain Annex.

- Hardware and Software – Hardware and software associated with the supported test domains and software interfaces required to build ATS were included.
- Signal Types – The scope was limited to digital, analog, Radio Frequency (RF), and microwave electrical signals.
- Testing Levels – The interface standards in the ATS Subdomain Annex are mandated for depot and intermediate level ATS only. The standards may be mandated for operational/organizational level use in the future.

The standards selected for inclusion in the ATS Subdomain Annex were found to be key for the generic open system architecture for ATS. The standards are based on commercial open system technology, have implementations available, and are strongly supported in the commercial marketplace. Standards in the ATS Subdomain Annex meet the following criteria:

- Availability **-** The standards are currently available.
- Commercial Acceptance **-** The standards are used by several different commercial concerns.
- Efficacy **-** The standards increase the interoperability of ATS hardware and software.
- Openness **-** Mandated standards are all open, commercial standards.

Standards that are commercially supported in the marketplace with validated implementations available in multiple vendors' mainstream commercial products took precedence over other standards. Publicly held standards were generally preferred. International or national industry standards were preferred over military or other government standards. Many standards have optional parts or parameters that can affect interoperability. In some cases, a standard may be further defined by a standards profile which requires certain options to be present to ensure proper operation and interoperability.

Previously, each of the Services had established their own sets of standards (e.g., technical architectures). The ATS Subdomain Annex is envisioned as a single generic open system architecture for ATS for the DoD. The ATS Subdomain Annex shall be used by anyone involved in the management, development, or acquisition of new or improved ATS within DoD. System developers shall use the ATS Subdomain Annex to ensure that new and upgraded ATS, and the interfaces to such systems, meet interoperability requirements. System integrators shall use this document to facilitate the integration of existing and new systems. Operational requirements developers shall be cognizant of the ATS Subdomain Annex in developing requirements and functional descriptions. ATS is a subdomain of the Combat Support domain of the JTA.

CS.ATS.1.5 TECHNICAL REFERENCE MODEL

CS.ATS.1.5.1 Hardware

The hardware interfaces in a typical ATS are shown in Figure CS.ATS-2. Mandates were only made for interfaces that have an impact on the interoperability and life-cycle costs of ATS across the DoD and for which widely accepted commercial standards exist. Mandates were not made for interfaces that are not supported by commercial standards, nor were they made for interfaces that do not affect the interoperability and life-cycle costs of DoD ATS. Unsupported interfaces that impact the interoperability and life-cycle costs of DoD ATS are identified in the section on emerging standards.

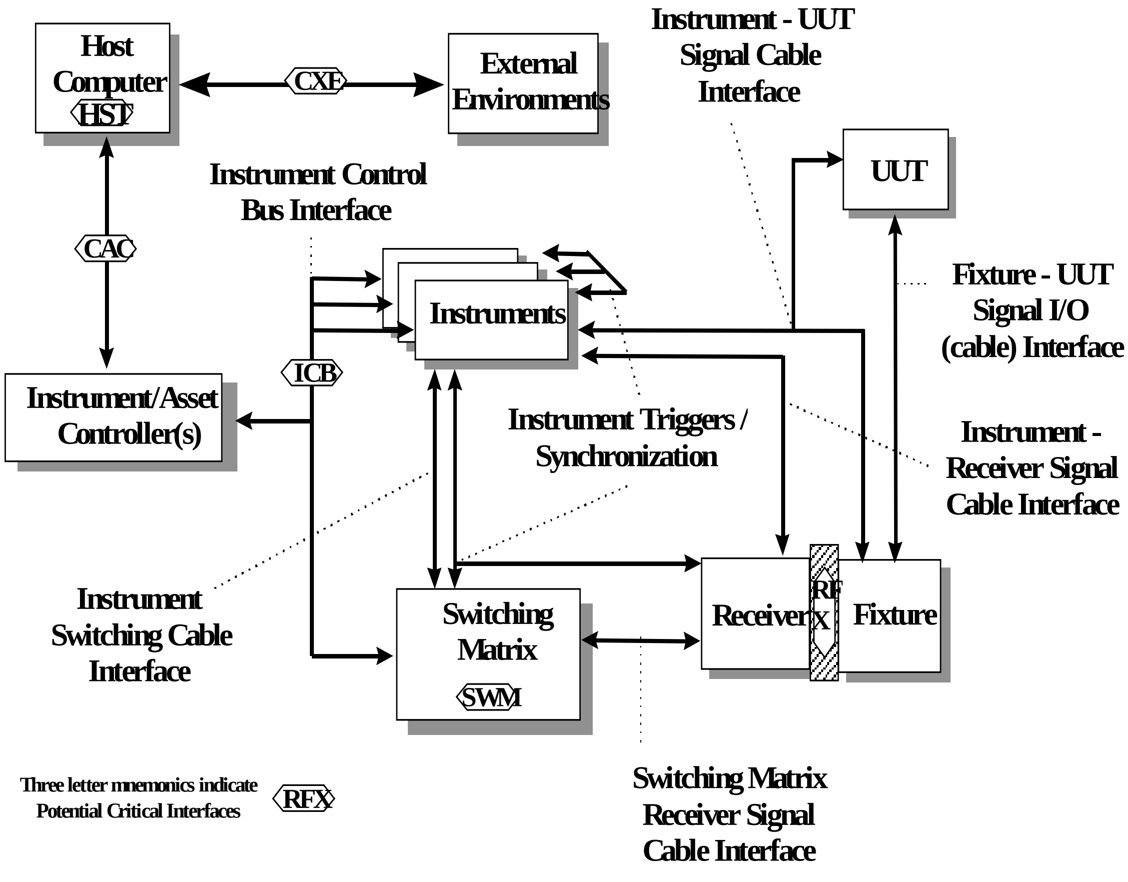

Figure CS.ATS-2 Hardware Interfaces

The interfaces shown in Figure CS.ATS-2 are listed alphabetically by mnemonic below:

- **Computer Asset Controller Interface (CAC)** describes the communication paths between the host computer and instrument controllers in a distributed system.
- **Computer to External Environments (CXE)** describes the communication methods between a host ATS and remote systems.
- **Host Computer Interface (HST)** describes the processing architecture of the primary control computer where the TPS is executed and through which the operator interfaces.
- **Instrument Control Bus (ICB)** interface describes the connection between the host computer or instrument controller and the test and measurement instruments in the ATS.
- **Receiver/Fixture Interface (RFX)** describes the interface between the receiver (part of the ATS) and the Fixture (part of the TPS). The RFX establishes an electrical and mechanical connection between the UUT and the ATS.

– **Switching Matrix Interface (SWM)** describes switch paths that connect ATS test and measurement instruments to pins on the RFX.

CS.ATS.1.5.2 Software

The software interfaces are introduced using two reference models, a run time view and a TPS development view. The interfaces applicable to the run time view are shown in Figure CS.ATS-3. These interfaces describe information processing flows as the TPS diagnoses a UUT. The TPS development interfaces are shown in Figure CS.ATS-4.

In these diagrams, Host Computer refers to computers that run the ATS and instrument asset controllers and computers that are subordinate to the host. The run time diagram presents a generic template for the functional organization of software processes. Subsets of this structure will appear on individual processors in a distributed-processing architecture. On any processor, if components shown on this diagram are present and interact, their interactions must comply with the interface requirements identified in this document.

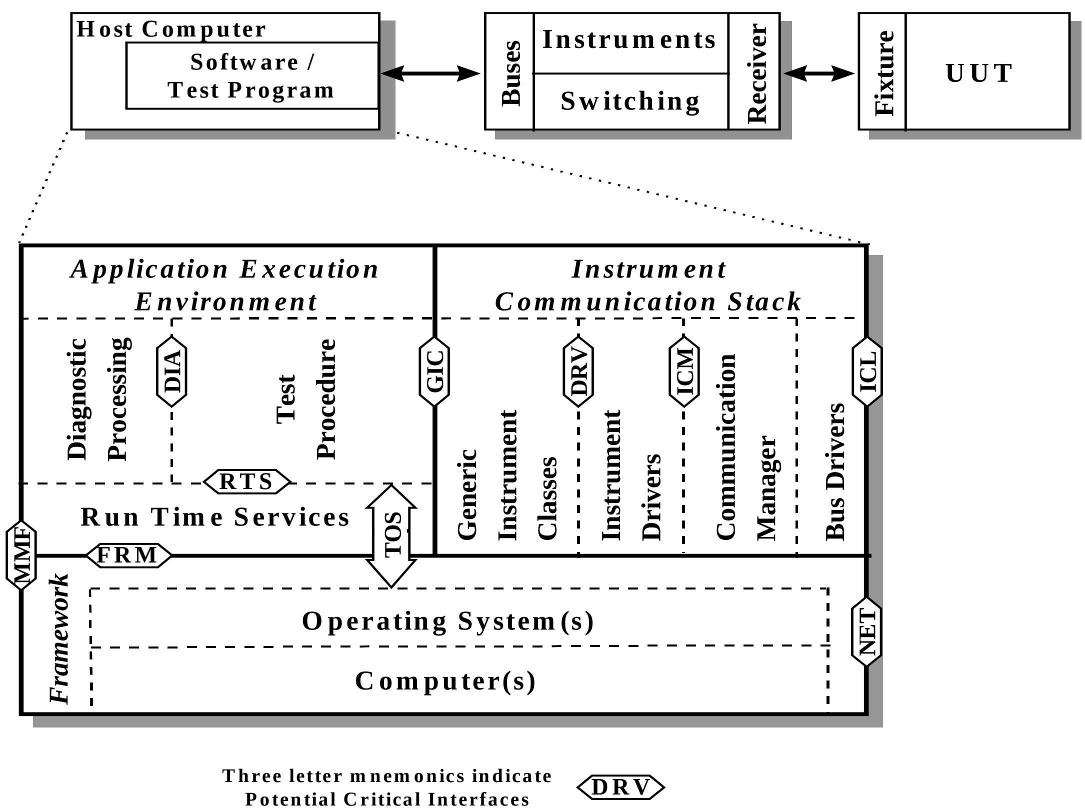

Figure CS.ATS-3 TPS Run Time Interfaces

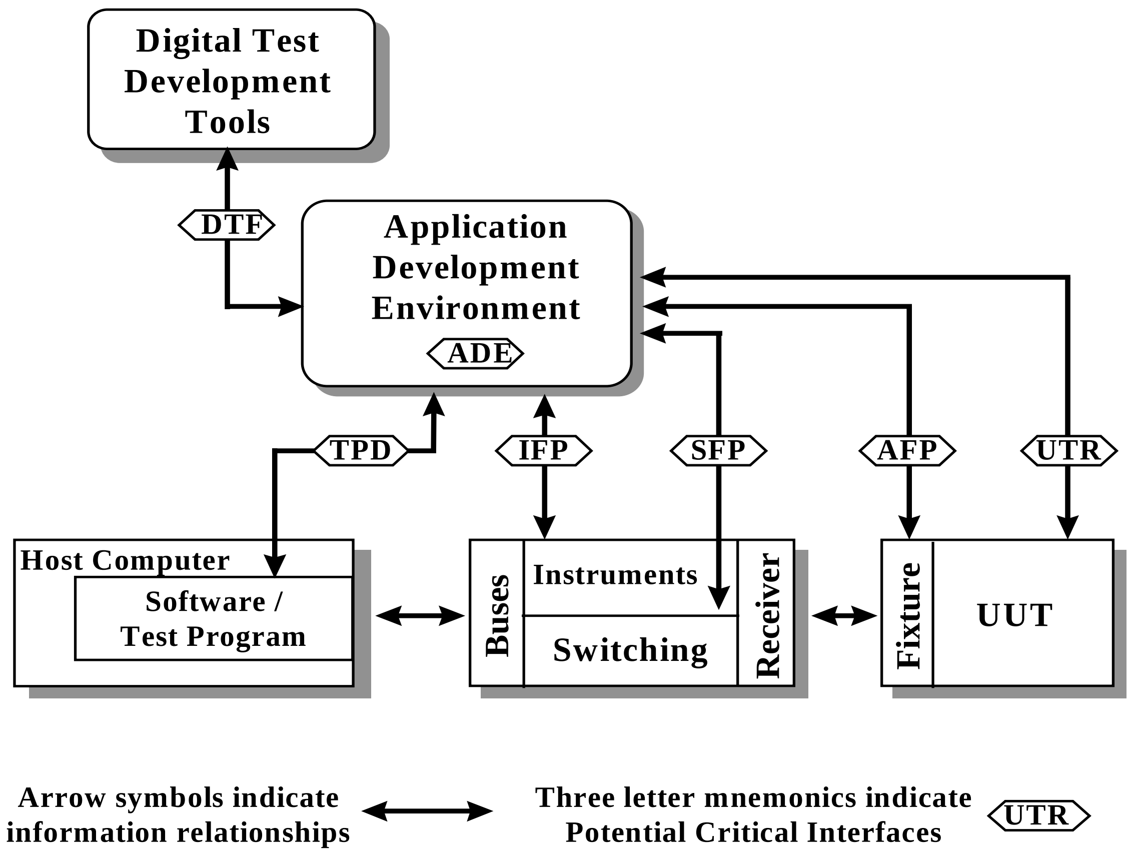

Figure CS.ATS-4 TPS Development Interfaces

The interfaces depicted in the run time view of Figure CS.ATS-3 are listed alphabetically by mnemonic below:

- **Diagnostic Processing (DIA)** is the interface protocol linking execution of a test with software diagnostic processes that analyze the significance of the test results and suggest conclusions or additional actions required.
- **Instrument Driver API (DRV)** is the Application Programming Interface (API) through which instrument drivers accept commands from and return results to Generic Instrument Classes.
- **Framework (FRM)** is a collection of system requirements, software protocols, and business rules (e.g., software installation) affecting the operation of test software with its host computer and Operating System (OS).
- **Generic Instrument Classes (GIC)** is the interface through which instrument drivers accept commands from and return results to test procedures or run time services serving the Test Program.
- **Instrument Command Language (ICL)** is the language in which instrument commands and results are expressed as they enter or leave the instrument.
- **Instrument Communication Manager (ICM)** is the interface between the instrument drivers and the Communication Manager that supports communication with instruments independent of the bus or other protocol used (e.g., VXI, IEEE-488.2, RS-232).
- **Multimedia Formats (MMF)** denotes the formats used to convey hyperlink text, audio, video and three-dimensional physical model information from multimedia authoring tools to the Application Development Environment (ADE), Application Execution Environment, and host framework.
- **Network Protocol (NET)** is the protocol used to communicate with external environments, possibly over a Local or Wide Area Network. The software protocol used on the CXE hardware interface is represented by the NET software interface.

- **Run Time Services (RTS)** denotes the services needed by a TPS not handled by the services supplied by the DRV, FRM, GIC, and NET, (e.g., error reporting, data logging).
- **Test Program to Operating System (TOS)** denotes system calls to the host OS made directly from the TPS.

The interfaces depicted in the development view of Figure CS.ATS-4 are listed alphabetically by mnemonic below:

- **Application Development Environments (ADE)** is the interface by which the test engineer creates and maintains a TPS, whether captured in the form of a text or graphical language.
- **Adapter Function and Parametric Data (AFP)** is the information and formats used to define to the ADE the capabilities of the test fixture, how the capabilities are accessed, and the associated performance parameters.
- **Instrument Function and Parametric Data (IFP)** is the information and formats used to define to the ADE the load, sense, and drive capabilities of the instruments, how these capabilities are accessed, and the associated performance parameters.
- **Switch Function and Parametric Data (SFP)** is the information and formats used to define to the ADE the interconnect capabilities of the switch matrix, how these capabilities are accessed, and associated performance parameters.
- **Test Program Documentation (TPD)** is human-understandable representations of information about the TPS for use by the TPS maintainer.
- **UUT Test Requirements (UTR)** is the information and formats used to define to the ADE the load, sense, and drive capabilities that must be applied to the UUT to test it, including the minimum performance required for a successful test.

CS.ATS.1.6 ANNEX ORGANIZATION

The ATS Subdomain Annex consists of three main sections. Section one contains the overview, section two contains the additions to the JTA core service areas for ATS, and section three contains the domain specific service areas for ATS. A list of sources is provided in Appendix B. In cases where the ATS Subdomain Annex does not address an interface to be used in an ATS, the JTA takes precedence. In cases where the JTA and ATS Subdomain Annex specify different standards for the same interface, the ATS Subdomain Annex takes precedence.

CS.ATS.1.7 CONFIGURATION MANAGEMENT

Configuration management of the ATS Subdomain Annex will be the responsibility of the joint service ARI. All changes will be approved by the ATS EAO with coordination from the ATS Management Board (AMB).

CS.ATS.2 ADDITIONS TO THE JTA CORE

CS.ATS.2.1 INTRODUCTION

The standards in the ATS Subdomain Annex apply in addition to the standards in the Combat Support Domain and the JTA core.

CS.ATS.2.2 INFORMATION PROCESSING STANDARDS

CS.ATS.2.2.1 Mandate Additions

CS.ATS.2.2.1.1 Instrument Driver API Standards

The DRV is the interface between the generic instrument class serving the test procedure and the instrument driver. The calls made available at this interface include calls oriented to software housekeeping, such as initializing the driver itself, and calls that cause the instrument to perform a function, such as arm and measure commands. The service requests crossing this interface are communications between generic ATS assets (e.g., digital multimeter) and specific ATS assets (e.g., vendor XYZ model 123 digital multimeter). The instruments are ATS assets, but the calls to the driver are either direct or close-to-direct consequences of action requests in the Test Procedure which is a TPS asset. Some instrument functions are available from a variety of instruments. However, the driver calls to access these functions vary from instrument to instrument. This interferes with TPS portability. Historically, cross-platform incompatibilities in the way drivers for the same instrument implement the same function have been a recurring ATS integration problem. In common commercial practice, the driver is acquired with the instrument from the instrument's original equipment manufacturer. The DRV API interface allows software developed by different organizations to work together. No standards are mandated in this version of the JTA, but an emerging standard is given in Section CS.ATS.2.2.2.1.

CS.ATS.2.2.1.2 Digital Test Data Formats

Digital Test Data Formats (DTF) describe the sequence of logic levels necessary to test a digital UUT. Digital test data is generally divided into four parts: patterns, timing, levels, and circuit models and component models that are used for the fault dictionary. In addition, certain diagnostic data may exist that are closely associated with the digital test data. This interface is intended to be used for capturing the output of digital automatic test pattern generators. No standards are mandated in this version of the JTA, but an emerging standard is given in section CS.ATS.2.2.2.2.

CS.ATS.2.2.2 Emerging Standards

CS.ATS.2.2.2.1 Instrument Driver API Standards

The following standard may be mandated in a future version of the JTA:

– VXI*plug&play* Systems Alliance Instrument Driver Functional Body Specification VPP-3.2, Revision 4.0, 2 February 1996.

CS.ATS.2.2.2.2 Digital Test Data Formats

A standard for describing DTF, known as LSRTAP, has become a de facto industry standard. The LSRTAP standard was submitted to the IEEE for formal standardization and is currently being voted on. The following standard may be mandated in a future version of the JTA:

– NAWCADLKE-MISC-05-PD-003, Navy Standard Digital Simulation Data Format (SDF), January 1998.

Note: The Navy specification for LSRTAP will be replaced with the IEEE standard (IEEE P1445) upon final approval from the IEEE.

CS.ATS.2.2.2.3 Generic Instrument Class Standards

The Generic Instrument Class (GIC) is the interface between the generic instrument classes serving the test procedure or run time services and the instrument driver. The service requests crossing this interface are communications between the TPS requirements (e.g., measure voltage of a sine wave) and generic ATS assets (e.g., digital multimeters, waveform generators, and power supplies). Industry has indicated an interest in pursuing a standard in this area. Some examples are the IEEE 1226 ABBET standard and the VXI*plug&play* Systems Alliance.

CS.ATS.2.2.2.4 Diagnostic Processing Standards

The diagnostic processing interface resides between the test procedure or run time services supporting the TPS and a diagnostic reasoner, diagnostic controller, or other diagnostic process. Diagnostic tools are most frequently encountered in one of three forms: expert systems, decision-tree systems and model-based reasoners. Other diagnostic tools are expert systems known as Fault Isolation System, and Expert Missile Maintenance Advisor; decision-tree systems including Weapon System Testability Analyzer, System Testability and Maintenance Program, System Testability Analysis Tool, and AUTOTEST; and model-based reasoners including Intelligent-Computer Aided Test, Portable Interactive Troubleshooter, Artificial Intelligence-Test, and Adaptive Diagnostic System.

Standardization in this area would allow tools to be written that can translate test strategy information to various test programming languages. Additionally, the tools would be interchangeable since one could use any tool to obtain the same output source code. Industry has indicated an interest in pursuing a standard in this area. One example is IEEE 1232.1: 1997, *Artificial Intelligence Exchange and Services Tie to All Test Environments* (AI-ESTATE).

CS.ATS.2.2.2.5 Adapter Function and Parametric Data Standards

This information defines the electrical behavior of the fixture which connects the UUT to the ATS. Functional descriptions are included to allow for the case of active fixtures. Describing the function of the fixture begins with a statement of the wirelist association between receiver terminals and UUT terminals. Performance parameters are required to complete the characterization of the path between the instrument and the UUT, so as to be able to construct a model of the effective instrument applied to the UUT signals (characterized with reference to the UUT interface). Industry has indicated an interest in pursuing a standard in this area. One example of this is the IEEE P1226.11 ABBET *Test Resource Information Model* (TRIM).

CS.ATS.2.2.2.6 ATS Instrument Function and Parametric Data Standards

This interface defines the capabilities of the ATS stimulus and measurement devices, how they are controlled, and how they are connected within the ATS. It includes:

- **Instrument Capabilities -** This defines what the instrument can measure, stimulate, and/or load the circuits to which it is attached. It includes identifying the function, such as measure volts, and quantitative performance characteristics including the range over which a voltage can be measured and the resolution and accuracy (as a function of choice of range) to be expected from the measured value.
- **Instrument Control -** The command vocabulary by which the instrument can be controlled to apply these behaviors.
- **Instrument Limits -** Limits are associated both with the safety of the instrument and surety of resolution performance. For example: *"Do not expose this instrument to more than 1 KV across the sensing terminals"* or *"Accuracy of voltage stimulus guaranteed with the instrument sourcing up to 100 mA."*

Industry has indicated an interest in pursuing a standard in this area. One example of this is the IEEE P1226.11 ABBET TRIM.

CS.ATS.2.2.2.7 ATS Switching Function and Parametric Data Standards

This interface defines the capabilities of the ATS switching devices, how they are controlled, and how they are interconnected with other ATS devices. It includes the possible states of the separately-setable switch elements, the connectivity through the switch in each such state, and electrical performance characteristics of the paths connected as a result of the switch state. The parametric information includes as-installed electrical path performance from the point to which the instrument characteristics are referenced out to the receiver/fixture disconnect surface. Industry has indicated an interest in pursuing a standard in this area. One example of this is the IEEE P1226.11 ABBET TRIM.

CS.ATS.2.2.2.8 UUT Test Requirements Data Standards

High re-host costs in the past have been associated with the failure to record or preserve the signal-oriented action capabilities *as required* as opposed to *as used.* This problem is most visible in the allocation phase of TPS development. When a TPS is transported or re-hosted, the resources requested by the TPS must be allocated to assets in the target ATS. This task would be simplified if UUT test requirements in the form of load specifications, measurement requirements, and stimuli requirements that must appear at the UUT interface were available. Industry has indicated an interest in pursuing a standard in this area. Some examples of this are the IEEE P1029.3 Test Requirements Specification Language (TRSL) and the Electronics Industry Association's Electronic Design Interchange Format (EDIF).

CS.ATS.2.2.2.9 TPS Documentation Standards

The TPS Documentation interface consists of the supporting documentation, provided by the TPS developer, whose purpose is to convey an understanding of the design philosophies incorporated into the various elements of the TPS hardware and software, along with detailed instructions for selected processes such as how to regenerate the executable program from the source libraries provided. These documents may include the Test Strategy Report (TSR), Diagnostic Flow Charts (DFC), Test Requirements Document (TRD), Test Diagrams, Test Program Instruction (TPI), and Automatic Test Program Generator (ATPG) support data. These data are bundled together in the Test Program Documentation (TPD) interface. The following Data Item Descriptions are being considered for mandates:

– DI-ATTS-80284A, Test Program Set Document.
– DI-ATTS-80285A, Engineering Support Data.

CS.ATS.2.3 INFORMATION TRANSFER STANDARDS

CS.ATS.2.3.1 Mandate Additions

CS.ATS.2.3.1.1 Data Networking Standards

In an ATS that has either internal (controller to controller) or external (controller to external host) networking, standardizing on a networking protocol should reduce the amount of time spent re-hosting a TPS between two organizations. This problem becomes more serious if the ADE that is controlling the ATS has built-in applications that are network objects (either clients, servers, routers, or other). In these instances, porting the ADE between platforms becomes more difficult since it may support different network protocols and different operating environments. Also important is the transfer of test result data for logistics and maintenance engineering purposes, i.e., tracking of UUT, failure modes, and test results analysis. By defining a specific protocol as the choice for data communications, these problems will be significantly reduced. Networking accelerates the distribution of updates for TPSs that are operational on a large number of widely distributed ATSs. No data networking standards are mandated in this version of the JTA, but an emerging standard is given in section CS.ATS.2.3.2.1.

CS.ATS.2.3.1.2 Instrument Communication Manager Standards

The ICM interface includes bus-specific options for communicating from the instrument driver to a supporting Input/Output (I/O) library. Until recently, vendors of IEEE-488 and VXI bus hardware provided software drivers for their buses that were different according to the hardware bus protocol or Operating System (OS) used. This situation interfered with the plug and play capabilities that users thought they were going to get from buying different instruments that all communicated by common hardware protocols. The same functions of the same instruments were not accessed through software in the same way across buses and host platforms. Different manufacturers of IEEE-488 cards had proprietary and unique software calls. Furthermore, Hewlett-Packard and National Instruments, the two leading vendors of VXI slot0 cards and embedded controllers, used different I/O calls to access instruments. This impeded the transporting of instrument drivers, ADEs, and test programs from one set of hardware to another. Without a standard ICM interface, vendors cannot provide interoperable or portable instrument drivers because different vendors would use different I/O drivers at the very lowest layer of the software. This forces instrument drivers to be tailored to specific I/O calls for each test station and lowers the likelihood that instrument drivers will be commercially available for each configuration. In addition, standard I/O software allows one to place parameters such as bus addresses and instrument addresses in the instrument driver instead of the test program. No instrument communication manager standards are mandated in this version of the JTA, but an emerging standard is given in Section CS.ATS.2.3.2.2.

CS.ATS.2.3.2 Emerging Standards

CS.ATS.2.3.2.1 Data Networking Standards

ATS and development systems that are elements of ATS must maintain networking capabilities that conform with current Internet standards. Current Internet standards are identified in the Internet Official Protocol Standards Index as released by the Internet Architecture Board (IAB), which may be mandated in a future version of the JTA:

– Any hardware that has support for the software protocol standards specified in JTA Section 2.3.2.1.1.2.1.1, Transmission Control Protocol (TCP) and JTA Section 2.3.2.1.1.2.1.3, Internet Protocol (IP) may be used; however TCP and IP are mandated by the JTA core document. Unacknowledged, connectionless, datagram transport services will not be used in ATS.

CS.ATS.2.3.2.2 Instrument Communication Manager Standards

A standard ICM interface enables higher level software to be interoperable and portable between vendors and across different platforms. This improves the interoperability of test software and the ability to re-host test software from one test system to another. The following specification may be mandated in a future version of the JTA:

– VXI*plug&play* (VPP) Systems Alliance Virtual Instrument Standard Architecture (VISA) Library, VPP-4.3, 22 January 1997.

CS.ATS.2.4 INFORMATION MODELING, METADATA, AND INFORMATION EXCHANGE STANDARDS

CS.ATS.2.4.1 Mandate Additions

There are currently no additions applicable to ATS with respect to Information Modeling, Metadata, and Information Transfer Standards as specified in Section 2.4 of the JTA.

CS.ATS.2.4.2 Emerging Standards

There are currently no emerging standards identified in this section of the ATS Subdomain Annex.

CS.ATS.2.5 HUMAN-COMPUTER INTERFACE STANDARDS

CS.ATS.2.5.1 Mandate Additions

There are currently no additions applicable to ATS with respect to Human-Computer Interface Standards as specified in Section 2.5 of the JTA.

CS.ATS.2.5.2 Emerging Standards

There are currently no emerging standards identified in this section of the ATS Subdomain Annex.

CS.ATS.2.6 INFORMATION SYSTEMS SECURITY STANDARDS

CS.ATS.2.6.1 Mandate Additions

There are currently no additions applicable to ATS with respect to Information Systems Security as specified in Section 2.6 of the JTA.

CS.ATS.2.6.2 Emerging Standards

There are currently no emerging standards identified in this section of the ATS Subdomain Annex.

CS.ATS.3 SUBDOMAIN SPECIFIC SERVICE AREAS

CS.ATS.3.1 SOFTWARE ENGINEERING SERVICES

CS.ATS.3.1.1 Mandates

CS.ATS.3.1.1.1 Test Program to Operating System Calls

The TOS interface defines calls to host OS functions from the TPS. Some TPSs are highly dependent upon system calls unique to the initial TPS development system OS. A common use of calls to the OS in a TPS is in the area of file I/O. At the time of re-host, the OS calls may not be supported on the target ATS. OS calls are a chronic cause of non-portability in software. The best measure that will alleviate the transportability and re-hostability problems associated with OS calls is to ban them entirely. This also ensures that the TPS is developed with an ADE that provides enough encapsulated run time services to preclude the need for direct calls to the OS. The problems associated with calling OS utilities from within a TPS can be generalized to problems that occur if the next interface in the process is bypassed. For example, interoperability will be reduced if an instrument driver bypasses the ICM interface and calls a function outside of the VISA (VPP-4.x) library or if functions that are supported by VISA are embedded in an instrument driver and implemented in a non-standard manner. No test program to operating system call standards are mandated in this version of the JTA, but a rule which may be mandated in a future version of the JTA is given in Section CS.ATS.3.1.2.1.

CS.ATS.3.1.2 Emerging Standards

CS.ATS.3.1.2.1 Test Program to Operating System Calls

The following rule may be mandated in a future version of the JTA.

– Any element of the technical architecture that is implemented shall not be bypassed by a direct communication to another interface or layer further on in the process.

CS.ATS.3.2 DATA/INFORMATION SERVICES

CS.ATS.3.2.1 Mandates

This version of the ATS Subdomain Annex does not contain any domain-specific mandated standards in the area of data/information services.

CS.ATS.3.2.2 Emerging Standards

CS.ATS.3.2.2.1 Run Time Services

The RTS interface encompasses data logging services, operator I/O, timing and tasking control, and resource allocation performed at execution. This interface defines the means by which run time services are called during TPS execution. Although standards do not exist, various implementations do. Standardization in this area would allow the use of various test executives with any language that they support. Proprietary implementations of the interface between the TPS and Test Executive exist. However, the means by which various run time services are called depends upon the implementation. Direct transportability of a TPS across platforms will be compromised if the TPS requires run time services that are not supported on both systems or if the calling method differs between the host and target platforms.

Industry has indicated an interest in pursuing a standard in this area. Some examples are IEEE P1226.10, Microsoft's COM/OLE *(Component Object Model/Object Linking and Embedding)*, and Object Management Group's CORBA *(Common Object Request Broker Architecture)*.

CS.ATS.3.3 PLATFORM/ENVIRONMENT SERVICES

CS.ATS.3.3.1 Mandates

CS.ATS.3.3.1.1 Computer to External Environments

The Computer to External Environments (CXE) interface describes the communication methods between a host ATS and remote systems. This includes paths between the target ATS host computer and other ATS systems as well as TPS development stations which are part of the Test Environment. This interface supports transporting TPS software and supporting documentation between organizations. Examples of this interface include Ethernet, RS-232, and IEEE-488.

Any hardware that has support for the software protocol standards specified in JTA Sections 2.3.2.1.1.2.1.1 and 2.3.2.1.1.2.1.3, Transmission Control Protocol (TCP) over Internet Protocol (IP), may be used.

CS.ATS.3.3.1.2 System Framework Standards

System frameworks provide a common interface for developers of software modules, ensuring that they are portable to other computers that conform to the specified framework. By defining system frameworks, suppliers can focus on developing programming tools and instrument drivers that can be used with any ADE that is compliant with the framework. System frameworks contain, but are not limited to, the following components:

- Compatible ADEs
- Instrument Drivers
- Operating System
- Required Documentation and Installation Support
- Requirements for the Control Computer Hardware
- Soft Front Panel
- VISA Interface and I/O Software

– VXI Instruments, VXI slot0, System Controller, VXI Mainframe

A system designed using a VXI*plug&play* system framework ensures that the ADE, DRV, GIC, ICM, and other FRM components are compatible and interoperable with each other. Following the system framework requirements also ensures that all necessary system components have been included, resulting in a complete and operational system. System frameworks increase the likelihood that ADEs will be available on multiple platforms, greatly enhancing the ability to move test software between platforms. While this does not ensure total portability of TPSs, it does eliminate the need to translate or rewrite the source code when it is ported. No system framework standards are mandated in this version of the JTA, but a standard which may be mandated in a future version of the JTA is given in Section CS.ATS.3.3.2.1.

CS.ATS.3.3.2 Emerging Standards

CS.ATS.3.3.2.1 System Framework Standards

The following standard may be mandated in a future version of the JTA.

– VXI*plug&play* System Alliance System Frameworks Specification, VPP-2, Revision 4.0, 29 January 1996.

CS.ATS.3.3.2.2 Receiver/Fixture Interface

The Receiver/Fixture (RFX) and generic pin map interfaces represent a central element of the ATS through which the majority of stimulus and measurement reach the UUT. Standardization of the RFX and pin map allows the same fixture to be used on multiple ATS. A standard pin map restricts the types of signals present at different positions on the receiver. Standardization of this interface increases the interoperability of test program sets, resulting in lower re-host costs. Industry has indicated an interest in pursuing a standard in this area. One example of this is the Receiver Fixture Interface (RFI) Alliance.

CS.ATS.3.3.2.3 Switching Matrix Interface

The Switching Matrix (SWM) interface and ATS receiver/fixture pin map represent a central element of the ATS for connecting ATS instrumentation to the UUT through a switch matrix. The SWM allows a variety of instruments to be connected to multifunction terminals identified by a standard receiver/fixture pin map. The combination of standardizing the SWM interface and a common receiver/fixture pin map gives the ATS the capability to accommodate any fixture that conforms to the pin map. Standardization of the SWM interface and receiver/fixture pin map increase interoperability by ensuring that ATS instruments needed to test a UUT can be switched to pins required by the fixture.

CS.ATS.3.3.3 Other Standards

The interfaces described in this section are provided for completeness of the ATS Subdomain Annex and to make readers aware that these interfaces have been addressed. Standards for these interfaces are not mandated because they were not found to be key for the generic open system architecture for ATS.

CS.ATS.3.3.3.1 Computer Asset Controller Interface

The Computer Asset Controller (CAC) interface describes the communication paths between the host computer and instrument controllers in a distributed system. These interfaces may be internal or external to the host computer. Examples of internal interfaces are Industry Standard Architecture (ISA) and Peripheral Component Interface (PCI). Examples of external interfaces are IEEE-488, RS-232, Ethernet, Multisystem Extension Interface, and Modular System Interface Bus.

CS.ATS.3.3.3.2 Host Computer Interface

The Host Computer (HST) interface describes the processing architecture of the primary control computer where the TPS is executed and through which the operator interfaces. Portions of the HST interface affect the interoperability of ATS. These requirements are included in the Frameworks software interface.

CS.ATS.3.3.3.3 Instrument Control Bus Interface

The Instrument Control Bus (ICB) interface describes the connection between the host computer or instrument controller and the test and measurement instruments in the ATS. Examples of these interfaces are IEEE-488, VME, and VME Extensions for Instrumentation (VXI).

CS.ATS.3.3.3.4 Instrument Command Language

The Instrument Command Language (ICL) interface describes how instrument commands and results are expressed as they enter or leave test and measurement instruments. The requirements for this interface are satisfied by the DRV and GIC interfaces.

CS.ATS.3.3.3.5 Application Development Environments

The Application Development Environment (ADE) interface describes how the test engineer creates and maintains a TPS, whether it is captured in the form of a text or graphical language. This interface was not mandated because the requirements for the ADE are restricted by the FRM interface.